

FIG. 1

200

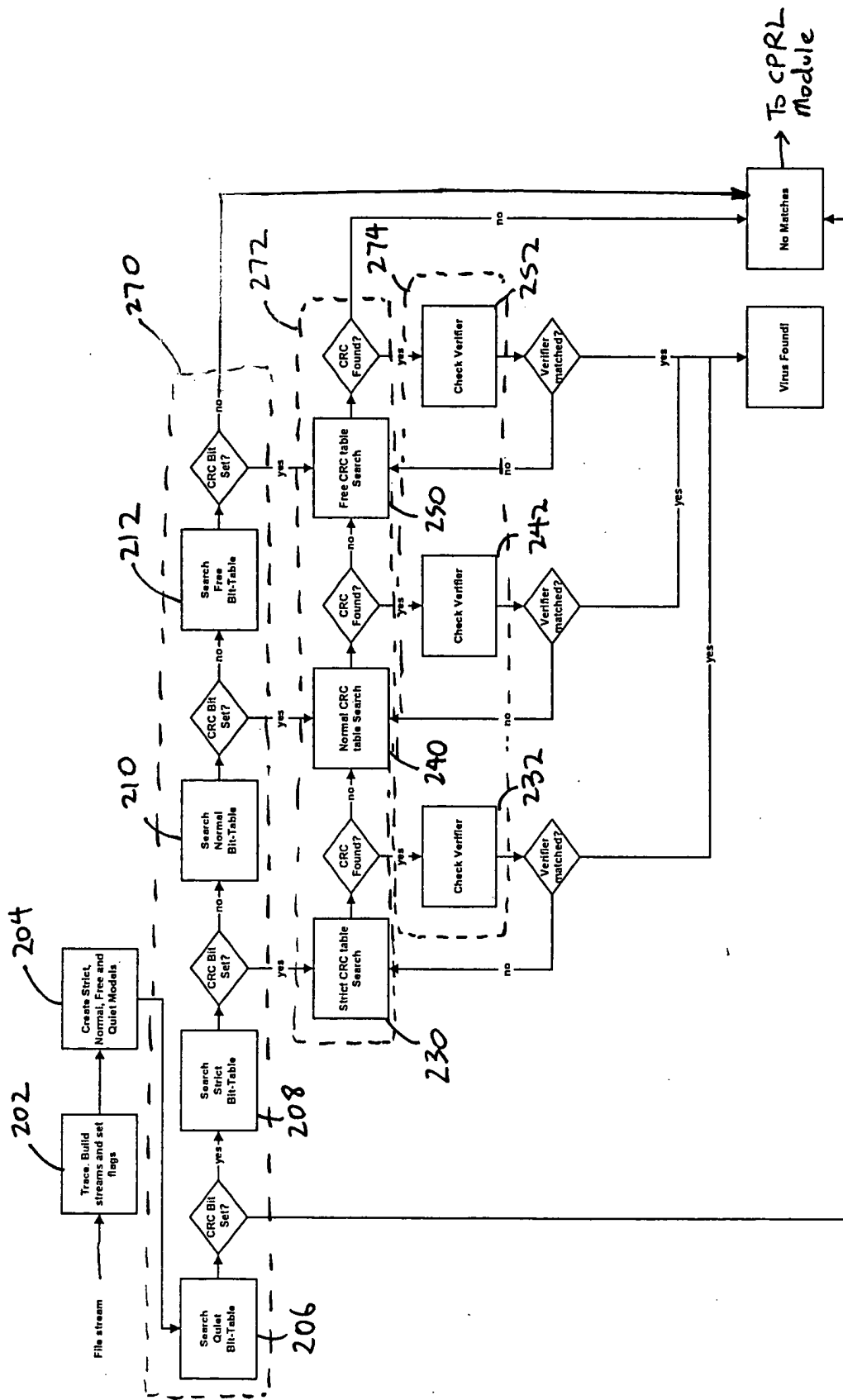


FIG. 2

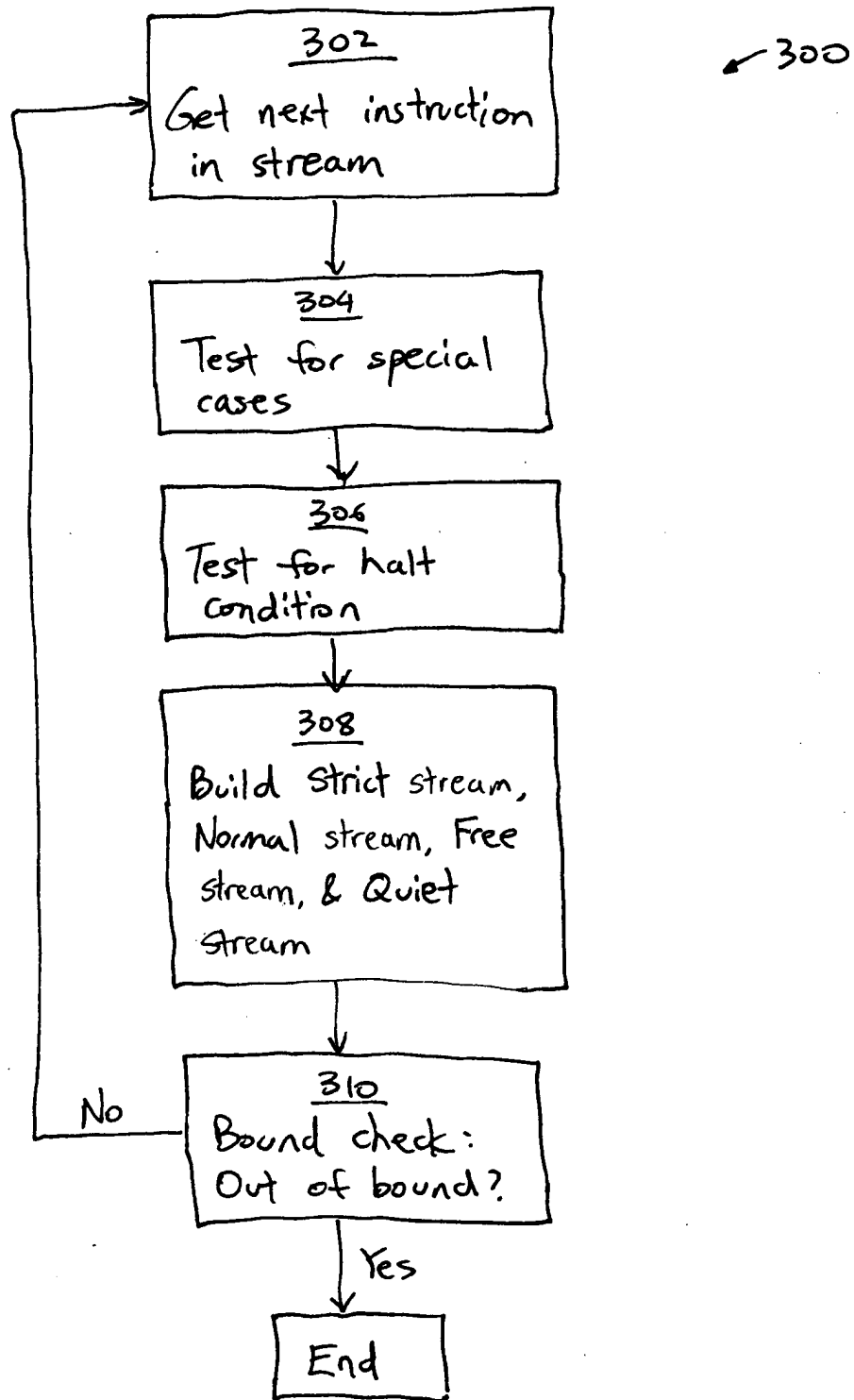


FIG. 3

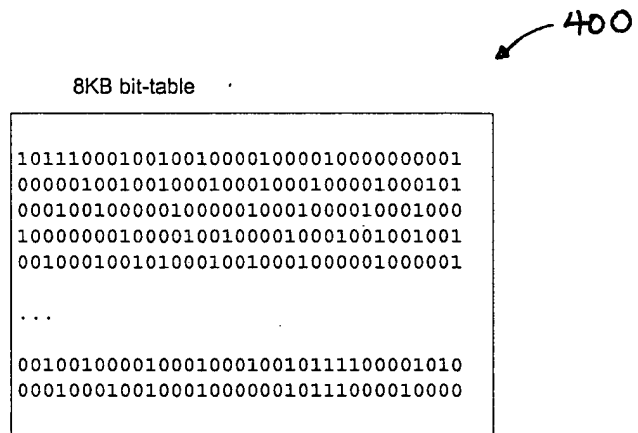


FIG. 4

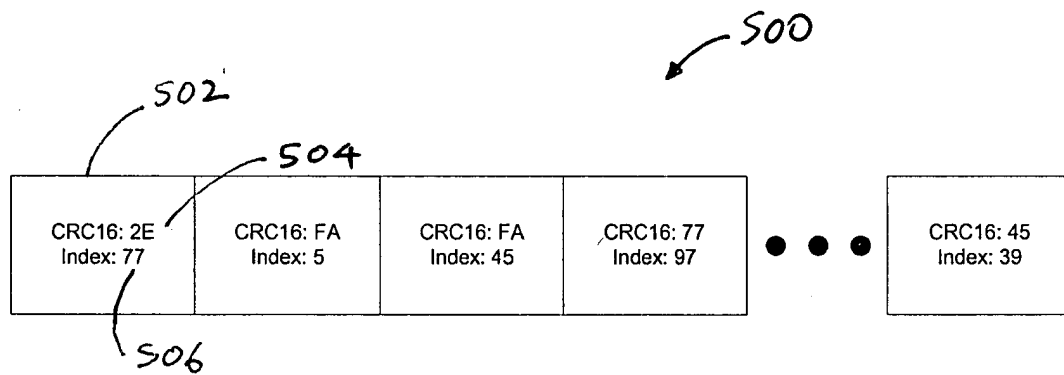


FIG. 5

Verifier Element

600

Verification tests: <crc, traceinfo, cpri>
CRC1: <crc used for verification>
CRC2: <a secondary crc used for verification as necessary>
crc1_len: <number of instructions represented by CRC1>
crc2_len: <number of instructions represented by CRC2>
crc1_model: <model represented by crc1>
crc2_model: <model represented by crc2>
TRACEINFO: <trace information that can be used as verifiers>
CPRL: <small cpri string used to verify information for this known virus>
Virus Name: <name of the virus this verifier represents>

602

FIG. 6

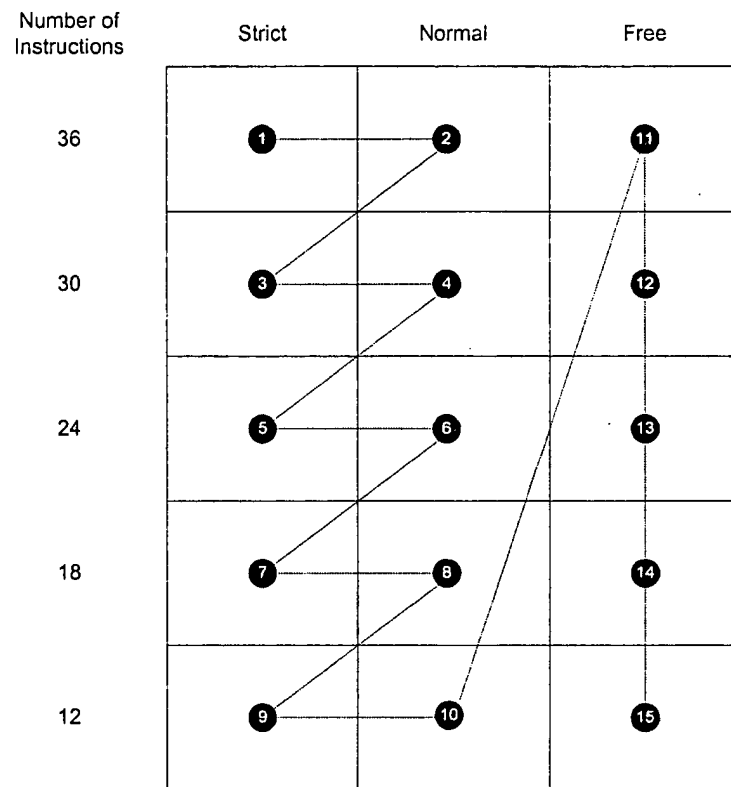


FIG. 7

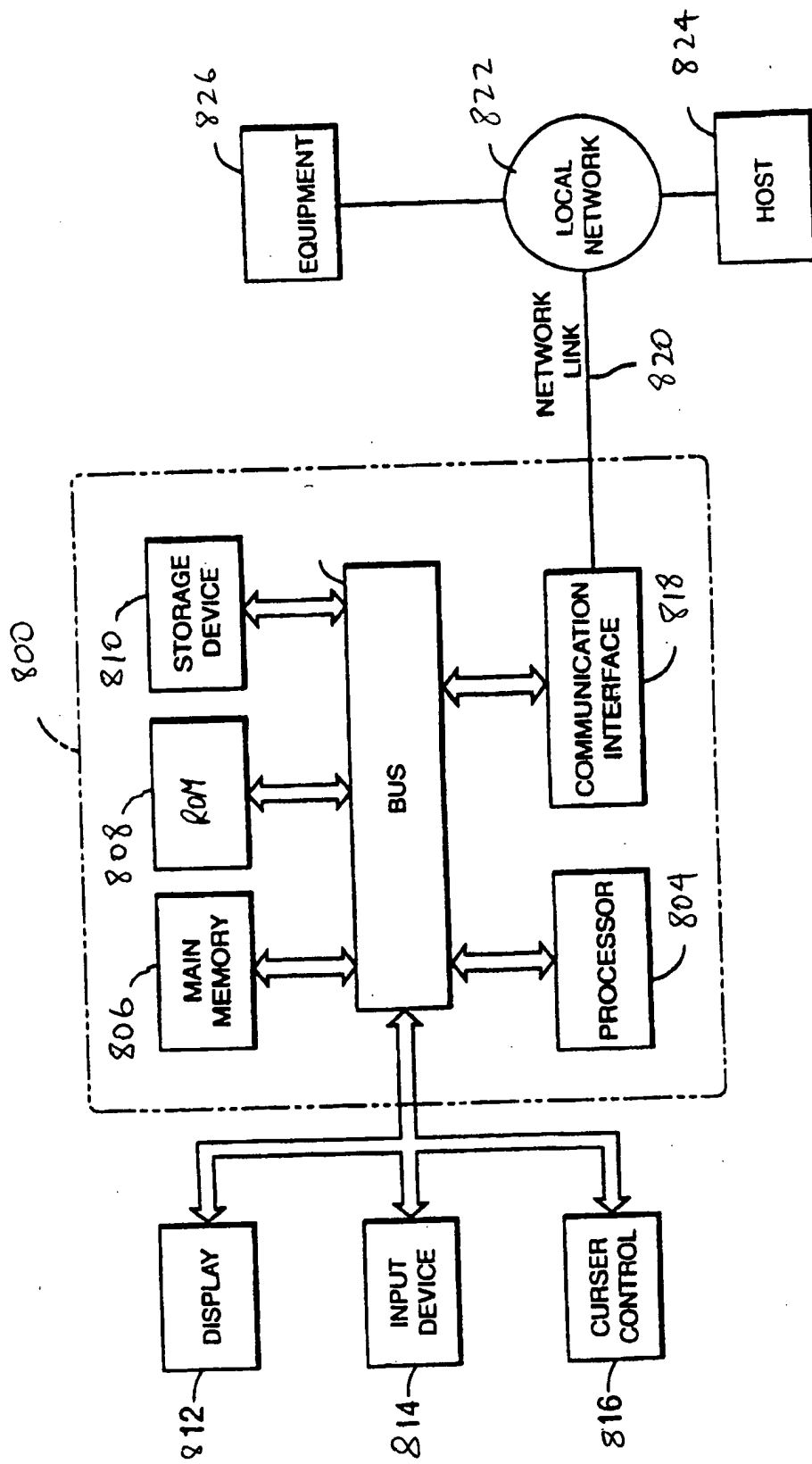


FIG. 8

Any two byte CRC contains a value from 0 to 65535. An eight kilobyte table contains 65536 bits. Therefore each bit in the 8k table can therefore represent a value from 0 (the first bit) to 65535 (the last bit). In other words, every value from 0 to 65531 can be represented by a single bit in an 8k table.

The following is an example of an algorithm that can be used to mask a CRC value to a single bit in the table:

C = CRC; B = Byte in table that contains the bit. b = the bit number that represents C. M = mask byte.

Note: Bit values 0 to 7 are represented by location, not by value.

Example:

1000000 = bit 0 set.

0000100 = bit 5 is set.

Algorithm

$b = C \wedge 8$

$B = C / 8$

Set high bit only in M (1000000)

Shift byte right by count b. (if b = 0 then no shift occurs)

Bitwise AND M against B.

If NOT zero then match.

Pseudo Code

```
boolean
MASK (WORD C,          // CRC
      BYTE *TABLE) // pointer to table
{
    WORD B;           // which byte in table
    BYTE b;           // bit count
    BYTE M;           // bit mask
    BYTE *N;          // pointer to byte in table

    M = 0x8000;       // initialize to binary 10000000

    B = C / 8;        // number of byte in table
    b = C ^ 8;        // remainder of C / 8
    N = TABLE + B;   // point to byte in table
    M = M >> b;       // shift right 0 to 7 bits

    if (M AND *N)     // AND Mask and Byte N in table
        return true; // bit is set
    return false;     // bit is not set.
}
```

FIG. 9